



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## DOPLNĚK K SOFTWARE PRO MOTORISTICKÉ ZÁVODY

ADD-ON TO MOTO-RACING SOFTWARE

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Dominik Schneiderka

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Radek Štohl, Ph.D.

BRNO 2018

# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Dominik Schneiderka

**ID:** 186180

**Ročník:** 3

**Akademický rok:** 2017/18

**NÁZEV TÉMATU:**

## Doplněk k softwaru pro motoristické závody

### POKYNY PRO VYPRACOVÁNÍ:

1. Popište software pro měření a zpracování průběhu závodu.
2. Na základě rozboru softwaru upravte aplikaci a pokuste se odstranit zásadní chyby
3. Vytvořte modul pro komunikaci s existující externí číslicovou zobrazovací jednotkou.
4. Vytvořte databázový model pro modul zobrazování informací o průběhu závodu pro paddock a komentátora.
5. Ověřte funkčnost svého řešení.

### DOPORUČENÁ LITERATURA:

Vývoj na platformě .NET [online]. Microsoft, 2016. Dostupné z:  
<https://msdn.microsoft.com/cs-cz/library/ms310243.aspx>

AGARWAL, Vidya Vrat a James HUDDLESTON. Databáze v C# 2008: průvodce programátora. Vyd. 1. Brno: Computer Press, 2009, 424 s. ISBN 978-80-251-2309-6.

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

**Termín zadání:** 5. 2. 2018

**Termín odevzdání:** 21.5.2018

**Vedoucí práce:** Ing. Radek Štohl, Ph.D.

**Konzultant:**



doc. Ing. Václav Jirsík, CSc.  
předseda oborové rady



### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## **Abstrakt**

Hlavním cílem této bakalářské práce bylo zprovoznění komunikace se zobrazovací jednotkou skrz rozhraní RS-485 (scoreboard) a vytvořit aplikaci pro hlasatele. Zabudování komunikace se zobrazovací jednotkou do existujícího programového řešení Regularity Rally a zároveň oprava zásadních chyb tohoto programu. Pro správné fungování bylo potřeba pozměnit existující databázový model. Veškeré programy byly vytvářeny ve vývojovém prostředí Microsoft Visual Studio 2017, za použití programovacího jazyka C#. Databázový model byl vyvíjen za použití MySQL.

## **Klíčová slova**

Závody pravidelnosti, MySQL, databáze, Visual Studio, c#, měření časů, časomíra

## **Abstract**

The main goal of this bachelor's thesis was to establish communication with scoreboard via RS-485 and develop application for reporter. Communication with scoreboard was added to existing program Regularity Rally, also main errors were corrected. To ensure correct behavior the existing database was slightly modified. All programs were developed in Microsoft Visual Studio 2017 and written in programming language C#. The database was developed via MySQL.

## **Keywords**

Regularity rally, MySQL, database, Visual Studio, c#, timekeeping, scoreboard

### **Bibliografická citace:**

SCHNEIDERKA, D. *Doplňk k softwaru pro motoristické závody*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 39 s. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D..

## **Prohlášení**

„Prohlašuji, že svou bakalářskou práci na téma Doplněk k softwaru pro motoristické závody jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **17. května 2018**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce Ing. Radkovi Štohlovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **17. května 2018**

.....  
podpis autora

# Seznam symbolů a zkratek

## Zkratky:

FEKT	...	Fakulta elektrotechniky a komunikačních technologií
VUT	...	Vysoké učení technické v Brně
PC	...	Personal Computer (osobní počítač)
USB	...	Universal Serial Bus (univerzální sériová sběrnice)
COM	...	Communication port (komunikační port)
DB	...	Databáze
DC	...	Direct Current (stejnoseměrné napětí)
PDF	...	Portable Document Format

# Obsah

1	Úvod.....	1
1.1	Měřicí jednotka (chronometr) .....	1
1.2	Zobrazovací jednotka (scoreboard) .....	2
2	Existující softwarové řešení .....	3
2.1	Regularity Rally .....	3
2.2	Popis programu .....	3
2.3	Spuštění .....	5
2.4	Možnosti aplikace .....	6
2.4.1	Měření časů .....	6
2.5	Zhodnocení .....	8
3	Databázový model .....	9
3.1	Přidané a upravené tabulky .....	10
3.1.1	Tabulky pro správu uživatelů .....	11
4	Komunikace mezi sw a scoreboardem .....	13
4.1	Oprava kabeláže .....	13
4.2	Komunikační rychlost, sekvence bytů .....	13
4.3	Zabudování komunikace do Regularity Rally .....	14
4.3.1	Metody třídy Scoreboard .....	14
5	oprava regularity rally .....	16
5.1	Úprava okna pro měření závodů .....	16
5.1.1	Původní vzhled a funkce okna .....	16
5.1.2	Nový vzhled a funkce .....	16
5.2	Úprava okna pro nastavení zařízení .....	19
5.2.1	Původní vzhled a funkce .....	19
5.2.2	Nový vzhled a funkce .....	20
6	Program pro komentátora a paddock .....	22
6.1	Funkce programu .....	22
6.2	Třídy v programu .....	24
6.2.1	Třída ConnectMethodsToDB .....	24
6.2.2	Třída GlobalData .....	25
6.2.3	Třída GlobalMethods .....	25
6.2.4	Třída MethodsForAppLogin .....	25
6.2.5	Třída MethodsForChooseRace .....	26
6.2.6	Třída MethodsForChooseTour .....	26
6.2.7	Třída Permission .....	26
6.3	Práce s databázovým modelem .....	26
6.4	Vzhled a funkce jednotlivých programových oken .....	28
6.4.1	Přihlášení do databáze .....	28
6.4.2	Přihlášení do aplikace .....	29
6.4.3	Hlavní programové okno .....	29
6.4.4	Okno pro výběr závodu a výběr turnaje .....	32
6.4.5	Detailní okna aplikace .....	32
7	závěr .....	34
	Literatura .....	35
	Seznam příloh .....	36
	Příloha 1 – Databázový model .....	37



Příloha 2 – Databázový model – správa uživatelů.....	38
Příloha 3 – CD/DVD.....	39

## Seznam obrázků

Obrázek 1 - TAG Heuer CP505 [3] .....	2
Obrázek 2 – Úvodní obrazovka RegularityRally.....	4
Obrázek 3 – Okno pro měření závodů .....	7
Obrázek 4 - Nový vzhled okna pro měření závodů .....	18
Obrázek 5 - Původní nastavení zařízení první podokno .....	19
Obrázek 6 - Původní nastavení zařízení druhé podokno .....	19
Obrázek 7 - Nové nastavení zařízení první podokno.....	21
Obrázek 8 - Nové nastavení zařízení druhé podokno .....	21
Obrázek 9 - Hlavní okno programu Regularity Rally.....	23
Obrázek 10 - Komunikace mezi metodami a databází .....	27
Obrázek 11 – Okno pro přihlášení se k databázovému serveru.....	28
Obrázek 12 – Okno pro nastavení databázového serveru.....	28
Obrázek 13 – Okno pro přihlášení se do aplikace .....	29
Obrázek 14 – Hlavní okno programu Regularity Rally Reporter s načtenými daty.....	31
Obrázek 15 – Okno pro výběr závodu .....	32
Obrázek 16 – Okno pro přidání poznámky.....	33
Obrázek 17 – Okno detail jezdce .....	33

## Seznam tabulek

Tabulka 1 - Seznam tabulek původního databázového modelu [3].....	9
Tabulka 2 - Seznam nově přidaných tabulek do databázového modelu.....	10
Tabulka 3 - Tabulka časů.....	10
Tabulka 4 - Poznámky u jezdců.....	11
Tabulka 5 - Uživatelé.....	11
Tabulka 6 - Role .....	11
Tabulka 7 - Práva.....	11
Tabulka 8 - Časové značky .....	12
Tabulka 9 - Přiřazení rolí k uživatelům .....	12
Tabulka 10 - Přiřazení práv k rolím.....	12
Tabulka 11 - Sekvence bytů pro komunikaci se SCOREBOARD .....	14
Tabulka 12 - Databázové tabulky využívané v Regularity Rally Reporter .....	26

# 1 ÚVOD

Tato bakalářská práce se zabývá zprovozněním komunikace se zobrazovací jednotkou (scoreboard) a zabudování komunikace do existující aplikace Regularity Rally. Dále byla vytvořena aplikace pro komentátora s názvem Regularity Rally Reporter. Pro zobrazení dat využívá aplikace databázového modelu vytvořeného pro aplikaci Regularity Rally.

Aplikace Regularity Rally byla vyvinuta Ing. Petrem Křížem [3]. Aplikace je zaměřená na měření závodů pravidelnosti a veškerá změřená data ukládá do databáze vytvořené pomocí MySQL. V první kapitole této práce je program podroben rozboru a analýze z hlediska nejzásadnějších chyb.

Lehkých změn se dočkal i databázový model, který je využíván pro ukládání dat. Databázový model byl upraven z důvodu využití této databáze i pro aplikaci určenou pro komentátora a paddock. Aplikace byla vytvářena ve vývojovém prostředí Microsoft Visual Studio 2017, za použití programovacího jazyka C#.

Do aplikace Regularity Rally byl také zabudován modul, který zprostředkovává komunikaci se zobrazovací jednotkou a zobrazení základních informací přihlížejícím divákům. Komunikace se zobrazovací jednotkou je popsána v závěru této práce.

## 1.1 Měřicí jednotka (chronometr)

Pro měření závodů byla použita měřicí jednotka TAG HEUER 505, zapůjčená Fakultou elektrotechniky a komunikačních technologií Vysokého učení technického v Brně.

Zařízení umožňuje měřit ve dvou režimech, manuálně a automaticky. V manuálním režimu jednotka vyhodnocuje závody sama, avšak nikoliv tak, jak by vyhovovalo kritériím závodů pravidelnosti. Vyhodnocení probíhá na základě odečtení času konce kola s časem začátku kola.

V automatickém režimu jednotka posílá data přes sériové rozhraní RS232 do nadřazeného zařízení (nejčastěji počítače), rychlost komunikace je 1200 baudů. V tomto zařízení potom probíhá vyhodnocení závodů (za pomoci vhodného softwarového vybavení).

Měřicí jednotka obsahuje také termotiskárnu, díky které je schopna okamžitě vytisknout výsledky závodu. Jednotku lze napájet dvěma způsoby, buďto pomocí baterií (čtyř AA baterií), nebo externím napájecím zdrojem DC 9 – 12 V. Chronometr má v sobě zabudované vyhřívání, které má své vlastní napájení, je potřeba DC zdroje 12 V s maximálním proudem 300 mA.



Obrázek 1 - TAG Heuer CP505 [3]

## 1.2 Zobrazovací jednotka (scoreboard)

Zobrazovací jednotka byla vyrobena a sestavena v rámci diplomové práce pana Ing. Róberta Hevese v roce 2007 [11], následně byla podrobena revizi a opravě zásadních chyb panem Ing. Jaroslavem Večeřou v rámci jeho bakalářské práce v roce 2011 [10]. Jednotka pro zobrazování údajů využívá elektromagnetické 7 segmentové zobrazovače (celkem osm, čtyři na každé straně). Pro fungování je potřeba napájení o velikosti napětí 12 V a minimální proud o velikosti 4 A. Napájet jednotku je možné buďto z laboratorního zdroje, nebo z externí baterie.

Jednotka má ověřenou komunikaci přes sériovou linku RS485 (využívána pro komunikaci se softwarem). Na základové desce jsou však osazená i rozhraní RS232 a USB, funkčnost těchto rozhraní ale nebyla ověřena. Mikroprocesor zobrazovací jednotky má v sobě naprogramované přijímání protokolu „TAG Heuer HL960/990“ (viz kapitola 4).

## 2 EXISTUJÍCÍ SOFTWAREVÉ ŘEŠENÍ

Na trhu sice existuje mnoho aplikací pro měření motoristických závodů, pro tuto práci se však nedají použít. Jedná se totiž o komerční aplikace a tím pádem nejsme schopni tyto programy modifikovat, ani jinak zasahovat do jejich kódu. Proto se autor této práce omezil pouze na aplikace vyvinuté v rámci diplomových prací vypracovaných na FEKT VUT v Brně.

Jak již bylo zmíněno v úvodu, tato práce se zabývá rozšířením existujícího softwarového řešení. Autor práce měl na výběr ze dvou aplikací [3], [12], rozhodl se pro rozšíření softwaru „Regularity Rally“, který byl vypracován v rámci diplomové práce Ing. Petrem Křížem. Níže bude tento software popsán.

### 2.1 Regularity Rally

Jak již název napovídá, tento program se zabývá měřením závodů pravidelnosti. Jedná se o speciální disciplínu okruhových závodů. U tohoto typu závodů jde závodníkům o zajetí kol s co nejmenší časovou odchylkou od času referenčního. Referenční čas stanovuje komisař závodu. Buďto je čas určen, nebo se závodníkovi přiřadí na základě zajetého kola.

Závody mohou být vyhodnocovány i na základě jiných kritérií, než je odchylka od referenčního času. V některých závodech se vyhodnocuje i co nejpřesnější dodržení průměrné rychlosti. Stanovení této rychlosti může být například podle prvního zajetého kola, nebo může být pevně dána.

Nejčastěji se tento typ závodu pořádá pro kategorii starších aut, popřípadě veteránů, a to z toho důvodu, že po autech není požadována vysoká maximální rychlost. Je to také jeden z důvodů, proč jsou tyto závody vhodné pro amatérské závodníky.

### 2.2 Popis programu

Program byl vytvořen ve vývojovém prostředí Microsoft Visual Studio 2015 pomocí programovacího jazyka C# a zvládá komunikaci s databázovým serverem.

Program je rozdělen na jednotlivá okna. Ke všem jednotlivým oknům lze přistupovat z hlavního okna programu (viz Obrázek 2). V dolní části je zobrazen přehled vybraného závodu, nebo turnaje. Zde lze vidět počet závodníků, podrobnosti o závodnících, měřicí zařízení (zařízení kterým je měřen závod) a výsledky, také se zde nachází tlačítko pro tisk (vytiskne do PDF souboru přehled). Vedle přehledu se nachází tlačítko pro měření časů, které se stane aktivním až po vybrání závodu ze seznamu závodů.



V horní části hlavního okna se nachází všechny ovládací prvky, které jsou rozděleny do sedmi „podoken“ (přihlášení, zobrazení závodů/turnajů, výběr závodu/turnaje, správa položek, tvorba a správa závodů, tvorba a správa turnajů, další funkce).

„Přihlášení“ slouží pro přihlášení k databázovému serveru. V aplikaci se dá pohybovat i bez přihlášení, ale jakmile bude uživatel chtít použít funkci, která pracuje s databází, bude k přihlášení vyzván.

„Zobrazení závodů/turnajů“ slouží pouze k přepínání mezi závody a turnaji. Mění se přehled ve spodní části hlavního okna. Při přepnutí na turnaje se v přehledu nezobrazuje karta zařízení.

V „podokně“ „správa položek“ lze přidávat a upravovat soutěžní vozy, závodníky, měřicí zařízení a firmu, která měření provádí. Také zde lze upravovat závodní kategorie.

V tvorbě a správě závodů, jak již název napovídá, uživatel může vytvářet a spravovat jednotlivé závody. Závodům může uživatel přiřadit měřicí zařízení, činovníky, nebo jednotlivé závodníky. Lze také vytvářet poznámky k závodům a přidávat jednotlivé soutěžící z jednotlivých kategorií. Aplikace umožňuje do závodů přidávat i jízdy a typ jízdy (volná jízda, kvalifikace, ostrý závod).

„Tvorba a správa turnajů“ nabízí vytváření nových turnajů, přidávání závodů do turnajů, pro jakou kategorii aut se bude turnaj konat, body v turnaji a závodníky.

Nabídka „další funkce“ nabízí zobrazení databáze, nastavení tisku do PDF souboru (lze nastavit font pro tisk), nastavení a test zařízení pro sériovou linku. V této verzi programu lze připojit měřicí zařízení jen za pomoci sériové linky. Program je možné otestovat pomocí „programových stopek“. Jedná se o stopky, které měří čas přímo v programu a není třeba připojení externího měřicího zařízení. Tyto stopky se chovají stejně jako klasické stopky, lze měřit čas, přidávat mezičasy (kola), zastavit měření času a vynulovat čas.

## 2.3 Spuštění

Pro úplnou funkčnost programu (i s databází) je třeba mít na PC nainstalovaný MySQL server a k tomu například MySQL Workbench (vše se dá stáhnout bezplatně z webových stránek MySQL [4]).

Nejprve je nutné spustit MySQL server, budeme-li spouštět měřicí aplikaci na stejném PC jako server, tak nastavíme lokální MySQL server. Tento server lze nastavit tak, že při prvotním spuštění MySQL serveru necháme všechny nastavení v defaultních hodnotách (primárně je vše nastaveno na lokální server).

Po spuštění serveru nahrajeme databázi do MySQL Workbench, k této databázi nastavíme připojení na server. Po úspěšném zadání hesla k serveru a „spuštění“ skriptu databáze je vše připraveno pro použití měřicí aplikace.

Aby byla aplikace plně funkční, musí uživatel přiřadit k MySQL serveru následující přihlašovací údaje: uživatelské jméno „root“ a heslo „123456“ (bez uvozovek). V kódu programu je toto jméno a heslo pevně nadefinováno pro přístup k tabulkám databáze, které načítají data do rolovacích seznamů v aplikaci. Bez vytvoření tohoto uživatele sice bude aplikace fungovat, ale velmi omezeně, některé záznamy z databáze totiž nebudou korektní a nebude přístup k celé databázi.



Při spuštění aplikace je tedy možné se přihlásit k databázi. V základním nastavení přihlášení není třeba nic měnit (pokud nebylo nic změněno při nastavování databázového serveru). Přihlášení je defaultně nastaveno na lokální server a název databáze která byla vytvořena pomocí MySQL skriptu databáze. Stačí tedy jen zadat správné přihlašovací jméno a heslo a kliknout na „přihlásit“.

## 2.4 Možnosti aplikace

Aplikace je určená pouze pro jeden typ závodů a tím jsou závody pravidelnosti. Sice by se tato aplikace dala použít i pro měření jiných typů závodů, ale vyhodnocování by již nebylo správné a uživatel by naměřené údaje musel vyhodnocovat sám. Tím pádem by se z programového nástroje staly pouze stopky.

V současné době umí program spolupracovat pouze s měřícím zařízením TAG HEUER 505 a to přes sériovou linku RS232. K tomuto zařízení se dá přesně přiřadit přes který COM port bude komunikovat, kolik má datových bitů, baud rate, stop bity, paritu a handshake. Jednotlivá měřící zařízení je rovněž možné přiřazovat k jednotlivým závodům.

Z hlavní obrazovky lze přistupovat ke všem položkám aplikace, každá položka se otevře ve svém samostatném okně. Hlavní obrazovka také zobrazuje přehled vybraných závodů, popřípadě turnajů a umí vytisknout požadované přehledy do PDF souboru. Z vybraného závodu, nebo turnaje lze zobrazit závodníky, kteří se závodů účastní, měřící zařízení a výsledky. U zobrazení výsledků lze vybrat z jaké kategorie chceme výsledky zobrazovat. První tři místa se zobrazí barevně, a to zlatě, stříbrně, nebo bronzově.

### 2.4.1 Měření časů

Jedná se o hlavní funkci programu. Do této části programu se lze dostat až po vybrání závodu v hlavním okně. Měříme tedy časy pro daný vybraný závod. Zobrazí se dvě velké tabulky, v jedné jsou načtení závodníci, kteří se účastní závodu a ve druhé tabulce jsou zobrazeny naměřené časy (viz Obrázek 3).

Uživateli je umožněno u závodníků měnit jejich stav (připraven, na trati, diskvalifikován apod.) a přiřazovat závodníkům referenční časy. Referenční čas lze zadat ručně, nebo vybrat dle již zjetého kola.

Pro jednotlivá zjetá kola uživatel může vkládat poznámky dle vlastního uvážení. Hned vedle pole pro poznámku najdeme položku „stav“, pomocí které lze nastavit stav kola (právě probíhá, ukončeno, přerušeno apod.). K jednotlivým kolům můžeme také přidat penalizaci. Kola s penalizací se zbarví červeně a penalizace se automaticky přičte k celkovému času. Jednotlivé časy kol lze také mazat. Bohužel u kol není napsané jméno závodníka, ale pouze jeho startovní číslo.

Není-li připojeno externí měřící zařízení, můžeme nastavit měření času programové. V tomto případě se aktivují programové stopky, které byly popsány výše. V případě měření času pomocí zařízení lze zařízení vybrat ze seznamu zařízení, který je přiřazen k danému závodů.

### Obrázek 3 – Okno pro měření závodů

## 2.5 Zhodnocení

Aplikace přináší zobrazení závodů, turnajů, jejich výsledků a parametrů. Z hlavního okna se dá přistupovat ke všem funkcím programu. Veškeré funkce programu jsou přehledně zpracovány a při používání nedochází ke zmatení, že by uživatel nevěděl, co zrovna dělá.

Někteří uživatelé však mohou být ohromeni počtem tlačítek, které se na hlavním okně programu nachází. Jako vylepšení programu by nebylo špatné schovat tato tlačítka do kontextového menu v záhlaví hlavního okna. Tímto krokem by se také zvětšil prostor pro zobrazované informace a aplikace by se tak stala přehlednější.

Existují zde různé nedostatky, například již při spuštění aplikace, kdy je třeba pro připojení do databáze nastavit přesné uživatelské jméno a heslo (není to nikde zmíněno). V případě zadávání dlouhých názvů turnajů se v „podokně“ „správa turnajů“ na kartě „body – turnaj“ překrývají funkční prvky právě s názvem turnaje a uživateli je znemožněno zadání počtu bodů za různá umístění.

Hlavním prvek aplikace je právě měření závodů, avšak tento režim je vyřešen trochu nešikovně. Proto, aby byl uživatel schopný změřit určitý čas, musí mít ze seznamu závodníků vybraného jednoho závodníka. V případě, že by došlo ke změření více časů najednou (dokončení kola více závodníky v krátkém časovém úseku), tak by se všechny naměřené časy přiřadily k právě vybranému závodníkovi. Časy se poté musí ručně upravit a přiřadit správným závodníkům. Také se může stát, že čas nebude vůbec zaznamenán, z důvodu nevybrání závodníka ze seznamu závodníků. Dále by se dalo okno pro měření časů opět zpřehlednit schováním tlačítek do kontextového menu v záhlaví aplikace.

Program je také psán tak, že většina logických operací se skrývá v obsluze tlačítek. Lepší řešení by bylo mít vytvořené třídy, které, v rámci svých metod, vykonávají dané logické operace a v obsluze tlačítka by se odvolávalo na jednotlivé metody příslušné třídy.

Oproti komerčním programům je aplikace určena pouze pro jeden účel a nabízí připojení pouze jednoho měřicího zařízení. Tento nedostatek je pochopitelný, jelikož na komerčních programech většinou pracují týmy vývojářů a na této aplikaci pracoval pouze jeden člověk.

Aplikace splňuje požadavky na měření a zpracování výsledků pro závody pravidelnosti. Dokonce umí pracovat i se vzdálenou databází.

### 3 DATABÁZOVÝ MODEL

Databázový model aplikace Regularity Rally je podrobně rozebrán v diplomové práci [3]. V této práci bude popsána pouze úprava databáze, kterou bylo třeba provést pro fungování aplikace určené pro komentátora a další zásahy do databáze, kterých bylo potřeba. Schématické znázornění databáze se nachází v příloze této bakalářské práce (viz PŘÍLOHA 1 – DATABÁZOVÝ MODEL).

Databáze je vytvořena pomocí databázového jazyka MySQL. Pro její spuštění je nezbytné mít nainstalovan potřebný software (popsáno v kapitole 2.3). MySQL byl vytvořen švédskou firmou MySQL AB a momentálně se jedná o jeden z nejrozšířenějších databázových systémů. Vysoké popularitě se těší mimo jiné i díky tomu, že je zcela zdarma a dostupný z webových stránek výrobce [4]. Aktuálně je MySQL vlastnictvím společnosti ORACLE.

**Tabulka 1 - Seznam tabulek původního databázového modelu [3]**

Název tabulky	Popis tabulky
prg_cars	Informace o vozidle
prg_cat	Informace o závodní kategorii
prg_cat_points	Rozdělení bodů v kategorii
prg_cat_races	Spojení kategorie se závodem
prg_cat_tour	Spojení kategorie s turnajem
prg_comp	Informace o soutěžících v závodě
prg_comp_cat_race	Přiřazení soutěžícího k závodní kategorii
prg_comp_cat_tour	Přiřazení soutěžícího k turnajové kategorii
prg_comp_tour	Soutěžící v turnaji
prg_device	Informace o měřicím zařízení
prg_inputs	Vstupy měřícího zařízení
prg_jury	Informace o činovnicích
prg_laps	Tabulka jednotlivých zajetých kol
prg_notes	Poznámky
prg_pass	Časy průjezdů kolem měřicích zařízení
prg_racers	Informace o jezdcích
prg_races	Informace o závodních podnicích
prg_races_device	Přiřazení měřícího zařízení k závodu
prg_races_tour	Přiřazení závodu do turnaje
prg_reference_lap	Referenční časy závodníků
prg_settings	Informace o společnostech zajišťujících měření a nastavení závodů
prg_split	Informace o mezechasech
prg_status	Tabulka statusů
prg_time	Tabulka se závodními časy
prg_tour_points	Rozdělení bodů v turnaji
prg_tour_ranks	Pořadí soutěžících v turnaji
prg_tours	Informace o turnaji

### 3.1 Přidané a upravené tabulky

Do databáze byly přidány datové tabulky pro správu uživatelů. Díky tomu datový model nyní umožňuje přidávat uživatele a jednotlivým uživatelům přiřazovat patřičná oprávnění a jejich role. Schématické znázornění databázových tabulek pro správu uživatelů se nachází v příloze této bakalářské práce (viz PŘÍLOHA 2 – DATABÁZOVÝ MODEL – SPRÁVA UŽIVATELŮ).

Mimo tuto úpravu byla do databázového modelu přidána tabulka sloužící k ukládání poznámek pro jednotlivé závodníky. Tabulka *prg\_time* byla rozšířena o jednu položku.

**Tabulka 2 - Seznam nově přidaných tabulek do databázového modelu**

Název tabulky	Popis tabulky
prg_racers_note	Uchovává poznámky k jednotlivým závodníkům
user	Přihlašovací údaje o uživateli
role	Názvy jednotlivých rolí
permission	Jednotlivá oprávnění
timestamp	Časová známka
role_user	Přiřazení rolí uživatelům
role_perm	Přiřazení oprávnění rolím

#### Tabulka *prg\_time*

Z důvodu úpravy aplikace Regularity Rally bylo potřeba sáhnout do tabulky *prg\_time* a přidat do ní ještě jednu hodnotu *time\_pass*. *Time\_pass*, slouží pro uchování času průjezdu (čas kdy byl závodník změřen měřícím zařízením, od začátku závodu, nikoliv čas kola).

**Tabulka 3 - Tabulka časů**

Název	Datový typ (délka)	Druh klíče	Popis
ID_time	int(11)	Primární	
ID_comp	int(11)	Cizí	Propojení s tabulkou <i>prg_comp</i>
ID_status	int(11)	Cizí	Propojení s tabulkou <i>prg_status</i>
ID_lap	int(11)	Cizí	Propojení s tabulkou <i>prg_laps</i>
Time	VARCHAR(20)		Naměřený čas jízdy
Pen	VARCHAR(20)		Penalizace času
Pen_notes	TEXT		Poznámka k penalizaci
Time_final	VARCHAR(20)		Celkový výsledný čas
notes	TEXT		Poznámka
Time_pass	VARCHAR(20)		Čas průjezdu

#### Tabulka *prg\_racers\_note*

Tabulka slouží pro záznam poznámek k jednotlivým závodníkům a je využívána pouze u programu pro komentátora. Za pomoci cizího klíče *ID\_racer* je provedeno propojení s tabulkou *prg\_racers*.

**Tabulka 4 - Poznámky u jezdců**

Název	Datový typ (délka)	Druh klíče	Popis
ID_note	int(11)	Primární	
ID_racer	int(11)	Cizí	Propojení s tabulkou prg_racers
note	TEXT		Poznámka

### 3.1.1 Tabulky pro správu uživatelů

#### Tabulka User

V této tabulce se uchovávají přihlašovací údaje uživatelů, jejich jméno a příjmení. Heslo je v databázi šifrováno pomocí hashovací funkce „password“, je tedy zajištěna alespoň základní ochrana hesla.

**Tabulka 5 - Uživatelé**

Název	Datový typ (délka)	Druh klíče	Popis
ID_user	int(11)	Primární	
Login	VARCHAR(50)		Přihlašovací jméno
Password	VARCHAR(50)		Heslo (šifrované)
Name	VARCHAR(50)		Jméno uživatele
Lastname	VARCHAR(50)		Příjmení uživatele

#### Tabulka Role

Obsahem této tabulky jsou názvy rolí, které mohou být přiřazeny jednotlivým uživatelům a právům.

**Tabulka 6 - Role**

Název	Datový typ (délka)	Druh klíče	Popis
ID_role	int(11)	Primární	
Role_name	VARCHAR(50)		Název role

#### Tabulka Permission

Tabulka obsahuje názvy práv, které mohou být přiděleny jednotlivým rolím.

**Tabulka 7 - Práva**

Název	Datový typ (délka)	Druh klíče	Popis
ID_role	int(11)	Primární	
Package	VARCHAR(50)		Jména práv

### Tabulka Timestamp

V tabulce se shromažďují časové údaje, které značí, kdy se daný uživatel přihlásil do aplikace pro komentátora. Tabulka je spojená s tabulkou *user* za pomoci cizího klíče *ID\_user*.

Tabulka 8 - Časové značky

Název	Datový typ (délka)	Druh klíče	Popis
ID_timestamp	int(11)	Primární	
ID_user	Int(11)	Cizí	Propojení s tabulkou <i>user</i>
Time	DATETIME		Čas přihlášení

### Tabulka Role\_user

Tabulka zajišťuje propojení mezi tabulkami *user* a *role*. Umožňuje přidělit libovolnému uživateli jeho roli. Tabulka obsahuje dva cizí klíče *ID\_user* a *ID\_role*, které zajišťují propojení s příslušnými tabulkami.

Tabulka 9 - Přiřazení rolí k uživatelům

Název	Datový typ (délka)	Druh klíče	Popis
ID_RU	int(11)	Primární	
ID_role	int(11)	Cizí	Propojení s tabulkou <i>role</i>
ID_user	int(11)	Cizí	Propojení s tabulkou <i>user</i>

### Tabulka Role\_perm

Podobně jako tabulka *role\_user* spojuje tabulka *role\_perm* jednotlivé role s jejich právy. V tabulce se opět nachází dva cizí klíče *ID\_role* a *ID\_perm*, které zajišťují propojení s tabulkami *role* a *permission*.

Tabulka 10 - Přiřazení práv k rolím

Název	Datový typ (délka)	Druh klíče	Popis
ID_RP	int(11)	Primární	
ID_role	int(11)	Cizí	Propojení s tabulkou <i>role</i>
ID_perm	int(11)	Cizí	Propojení s tabulkou <i>permission</i>

## **4 KOMUNIKACE MEZI SW A SCOREBOARD**

Komunikace mezi PC a zobrazovací jednotkou je zabezpečená pomocí sériové linky RS-485. Jelikož většina PC již neobsahuje COM port, bylo využito převodníku USB na RS-485. Tyto převodníky se chovají podobně jako karty s COM porty připojené přímo na základovou desku počítače. Po připojení převodníku do USB portu a nainstalování příslušných ovladačů se v PC objeví COM port, který je plně připraven k práci.

### **4.1 Oprava kabeláže**

Vzhledem ke stáří zobrazovací jednotky bylo nutné nejprve opravit vodiče pro komunikaci přes RS-485 a napájecí vodiče.

U komunikačních vodičů byl jeden vodič upadlý ze svorkovnice a další vodiče se nacházely ve špatném stavu (polámané, odhalená živá část). Všechny vodiče (na obou koncích) byly pomocí krimpovacích kleští opatřeny novými dutinkami a podle schématu správně připojeny. Komunikační vodiče byly popsány, aby nedošlo při další manipulaci ke špatnému zapojení. Popsána byla i připojovací svorkovnice na desce plošných spojů.

Napájecí vodiče se rovněž nacházely ve špatném stavu a neměly na sobě banánky pro připojení ke zdroji. Vzhledem k tomu, že se jedná o lankové vodiče, tak oba napájecí vodiče byly opatřeny dutinkami. Jelikož je zobrazovací jednotka schopna pracovat i na baterii, jsou napájecí vodiče opatřeny svorkovnicí, aby se mohl snadno měnit zdroj napájení. Po přidání dutinek na konce napájecích vodičů se na vodiče namontovaly banánky černé a červené barvy pro jednoznačné označení kladného a záporného pólu.

### **4.2 Komunikační rychlost, sekvence bytů**

Před samotnou komunikací bylo třeba zjistit jakou rychlostí se se zobrazovací jednotkou má komunikovat. Mikroprocesor časomíry je naprogramovaný tak, aby uměl komunikovat s komunikačním protokolem „TAG Heuer HL960/990“ [2]. Po vyhledání tohoto protokolu na internetu bylo zjištěno, že se má komunikovat rychlostí 9 600 Baudů, datových bitů je 8, žádná parita a jeden stop bit.

Sekvence bytů se dá rovněž nalézt na internetu k danému protokolu, avšak pro přesnější pochopení a ověření správnosti bylo využito programu Circuit Pro od společnosti Tag Heuer (Vola Timing). V tomto programu se nastaví příslušný COM port, na kterém se nachází časomíra. Po úspěšném nastavení program začne posílat čas na časomíru. Dále bylo využito shareware programu Device Monitoring Studio [9] (zdarma na prvních 14 dní), díky kterému je možné monitorovat COM port a data, která se na COM port posílají. Byla zjištěna následující sekvence bytů:



**Tabulka 11 - Sekvence bytů pro komunikaci se SCOREBOARD**

0x0A	0x02	0x4C	0x30	0x09	0x41	data	0x09
LF	STX	display		HT	A		HT

**LF** – Line Feed (posun o řádek)

**STX** – Start of Text (začátek textu)

**Display** – Nastavení na jaký řádek se bude zapisovat (máme jen jeden řádek, neměníme)

**HT** – Horizontal Tab (horizontální tab)

**A** – hexadecimálně zapsané písmeno A

**Data** – jedná se o sekvenci 9 bytů.

Jelikož SCOREBOARD obsahuje pouze čtyři 7 segmentové zobrazovací prvky, tak z devíti datových bytů nás zajímají pouze poslední čtyři hodnoty (znaky které jsme schopni zobrazit).

### 4.3 Zabudování komunikace do Regularity Rally

Při vytváření modulu pro komunikaci se zobrazovací jednotkou bylo vycházeno z programu, který byl vytvořen v rámci semestrální práce. Program byl funkční pro komunikaci s jednotkou, avšak byl psán jako program pro příkazovou řádku. Bylo tedy zapotřebí program přepracovat, aby dokázal pracovat v grafickém prostředí. Rozšíření o komunikační modul se zobrazovací jednotkou bylo do Regularity Rally zabudováno jako nová třída s názvem **Scoreboard**.

Pro komunikaci se zobrazovací jednotkou slouží okno „Nastavení zařízení“ (viz kapitola 5.2). Zobrazovací jednotka se dá ovládat také z okna pro měření závodů (viz kapitola 5.1). Na hlavní obrazovce se nachází pouze tlačítko pro zobrazení času.

Aplikace Regularity Rally obsahuje nejružnější chybové hlášky ohledně zobrazovací jednotky. Kdykoliv se bude uživatel snažit poslat data na zobrazovací jednotku a nemá nastavený COM port pro připojení, je vyzván k jeho nastavení. V případě, že by došlo při přenosu informací k nějaké chybě, je na to uživatel upozorněn.

Bohužel zobrazovací jednotka neumí odesílat zprávy zpět do počítače, takže v aplikaci není možné hlídat, zda data přišla ve správném formátu. Z tohoto důvodu jsou všude, kde se dá komunikovat se zobrazovací jednotkou (kromě hlavního okna), přidány položky pro kontrolu poslaných dat (viz obdélník 2 v Obrázek 4). Takto si může uživatel aplikace zkontrolovat, alespoň vizuálně, zda se přenos povedl.

#### 4.3.1 Metody třídy Scoreboard

Třída v sobě obsahuje několik metod pro obsluhu zobrazovací jednotky. Mezi hlavní metody patří *DisplayAtSC*, *Send\_to\_COM*, *convert\_string* a *Print*.

Metoda *DisplayAtSC* je nejčastěji volanou metodou ve třídě. Metoda má také tři přetížení, při volání jakékoliv verze metody však získáme kýžený výsledek, a to je zobrazení dat na zobrazovací jednotce. V případě, že metodě nepošleme žádná data, zobrazí stejné údaje, které byly naposledy zobrazeny, ale pouze v případě, že jsme mezitím nezměnili sekvenci posílaných bytů. Dále můžeme metodě *DisplayAtSC* předat i data, ve formě textového řetězce, která chceme zobrazit. Poslední možností je předat

metodě objekt „TextBox“, což zajistí vypsání naposledy poslaných dat do pole pro kontrolu poslaných bytů.

Metoda *DisplayAtSC* se v každém případě odvolává na metodu *Send\_to\_COM*. Jak již název napovídá, tato metoda má za úkol poslat sekvenci bytů zobrazovací jednotce. Metoda zajišťuje přímou komunikaci se zobrazovací jednotkou.

Pomocí metody *convert\_string* jsme schopni jakýkoliv textový řetězec převést na jednotlivé byty, které se následně mohou poslat zobrazovací jednotce. Jediné omezení je zde v délce textového řetězce, ten může mít délku pouze čtyři znaky, protože jednotka nedokáže zobrazit více znaků. Uživateli je tedy vždy znemožněno zadat více jak čtyři znaky.

Poslední hlavní metodou je metoda *Print*. Metoda zajišťuje vytisknutí poslaných dat do „okénka“ pro kontrolu poslaných dat, které je předem nadefinované. Pomocí této metody je uživateli umožněno kontrolovat poslané znaky a analyzovat tak chybu v případě, že se něco porouchalo.

Celková posloupnost metod pro zobrazení údajů na zobrazovací jednotce je následující: *DisplayAtSC* -> *convert\_string* -> *Send\_to\_COM* -> *Print*.

Pro zobrazení času počítače třída využívá speciální metody *Time*, která zjistí aktuální čas. Čas poté převede na sekvenci bytů, pošle je na zobrazovací jednotku a zároveň do „okénka“ pro kontrolu poslaných dat. Jedná se o téměř stejnou posloupnost metod, jako je popsána výše. V posloupnosti metod je místo *DisplayAtSC* použita metoda *Time* a namísto *convert\_string* je použita metoda *convert\_time*, zbytek posloupnosti metod je stejný.

Vzhledem k tomu, že okno nastavení zařízení (viz kapitola 5.2) nabízí otestování připojení zobrazovací jednotky, tak se ve třídě **Scoreboard** nachází i metoda pro testování. Jedná se o metodu *Test*, pomocí které se na zobrazovací jednotku posílá posloupnost číslic od jedné do devítky.

Třída dále obsahuje několik dalších a jednodušších metod, které můžeme považovat za režijní metody. Jedná se například o metodu *set\_port*, která nastaví zvolený COM port z okna pro nastavení zařízení (viz kapitola 5.2). Další režijní metodou je metoda *State*, která vrací buďto logickou jedničku, nebo nulu, na základě toho, zda je právě poslán nějaký údaj na zobrazovací jednotku, či nikoliv.

## 5 OPRAVA REGULARITY RALLY

V následující kapitole budou rozepsány opravy a úpravy programu Regularity Rally od původní verze. Hlavním cílem bylo nalézt nejzásadnější chyby aplikace, které znemožňovaly kvalitní práci s měřicím programem.

Opravy byly prováděny na obrazovce pro měření závodů a v okně pro nastavení externích zařízení, které komunikují přes sériový COM port. Rozšíření programu o komunikaci se zobrazovací jednotkou je rozepsáno v kapitole 4.

### 5.1 Úprava okna pro měření závodů

Úprava ve své podstatě vychází ze zhodnocení aplikace Regularity Rally (viz kapitola 2.5), kde byl zjištěn nedostatek s měřením. Měření nebylo uzpůsobeno případům, kdy by bylo v krátkém časovém úseku změřeno více závodníků.

Dále se v tomto okně lehce pozměnily zobrazovaná data v přehledu závodníků. Aplikace nově začala měřit čas průjezdu (čas kdy byl závodník změřen měřicím zařízením, od začátku závodu). Do této obrazovky byly také zabudovány ovládací prvky pro časomíru (viz kapitola 4).

#### 5.1.1 Původní vzhled a funkce okna

Okno pro měření závodů obsahuje dvě tabulky a je tak pomyslně rozděleno na dvě části (viz Obrázek 3). Levá tabulka obsahuje seznam závodníků, kteří v závodě závodí a zobrazuje základní informace o jejich postupu závodem, včetně referenčního času. Pravá tabulka obsahuje záznamy zajetých kol, čas kola, číslo kola, penalizaci a poznámku ke kolu.

Pod tabulkami se nacházejí ovládací tlačítka, která ovládají právě tabulku nad sebou (například se pod oběma tabulkami nachází tlačítko „Tisk“, které slouží k tisku záznamů z tabulky nad tlačítkem).

Pod levou tabulkou je i „měřicí“ část aplikace, kde si uživatel může zvolit, zda chce měřit pomocí externího zařízení, nebo použije programové stopky. V případě měření pomocí externího zařízení je třeba ještě zvolit příslušné zařízení z tabulky „Zařízení“.

Na pravé straně se pod tabulkou nacházejí pouze tlačítka pro úpravu odjetých kol (přidání penalizace, nebo poznámky ke kolu). Aby byl uživatel schopen přidat penalizaci, nebo poznámku, tak musí mít vybrané kolo v tabulce kol.

#### 5.1.2 Nový vzhled a funkce

Pomyslné rozdělení okna na dvě části bylo zachováno (viz Obrázek 4). Na levé straně byla ubrána tabulka pro výběr měřicího zařízení, tato funkce je nyní implementována jiným způsobem (viz kapitola 5.2). Levá tabulka (tabulka pro zobrazení závodníků v závodě) byla rozšířena o nové položky, konkrétně se jedná o pořadí závodníků v závodě a celkový čas závodníka (viz číslo 1 v Obrázek 4). Pořadí je vypočítáváno na základě odjetých kol, což umožní časoměřiči závodu mít lehký přehled o tom, kdo pravděpodobně přijede jako další. Pod levou tabulku bylo přidáno i ovládání časomíry

(viz obdélník 2 Obrázek 4). Časoměřič si může vybrat různé módy, co bude zobrazovací jednotka zobrazovat. Může se jednat o libovolný vlastní údaj, počet kol do konce závodu (odvozeno podle prvního závodníka, s možností zobrazení i startovního čísla prvního závodníka), odpočet času (z předem nastaveného času), nebo aktuální čas (čas počítače, na kterém je aplikace spuštěna). Veškeré poslané údaje na zobrazovací jednotku se zobrazují v okně v levé části aplikace „Údaje odeslané na časomíru:“.

Měření časů je zjednodušeno. Nyní lze započít měření závodu v případě, že byl nastaven COM port pro měřicí jednotku. Není již potřeba vybírat zařízení pro měření ze seznamu.

Na pravé straně obrazovky, pod tabulkou s odjetými koly je, nově, tabulka, která shromažďuje nově naměřené časy pomocí měřicí jednotky (programové, či hardwarové; viz obdélník 3 v Obrázek 4). V této tabulce uživatel přiřazuje naměřeným časům startovní čísla závodníků, které potvrdí stisknutím klávesy „enter“. Po potvrzení času se čas uloží do databáze a připsíže do tabulky výše. Přes tlačítko „poznámka“ je možné přidat poznámku ke kolu. Tlačítko „vymazat“ čas odstraní.

V pravé tabulce (tabulka se záznamy kol) přibyl záznam o průjezdu, došlo také ke změně pořadí sloupců (viz obdélník 4 v Obrázek 4).

Změna způsobu měření časů byla provedena z toho důvodu, aby se měření urychlilo. Měření časů je kritická operace, kdy je třeba rychle naměřit více časů najednou. Toto řešení umožňuje změřit závodníky rychleji, avšak není to stále ideální řešení.

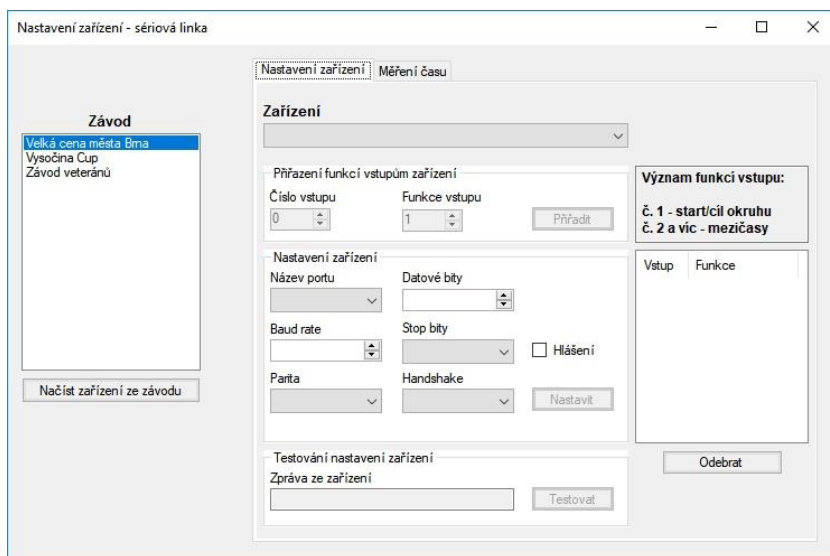
#### Obrázek 4 - Nový vzhled okna pro měření závodů

## 5.2 Úprava okna pro nastavení zařízení

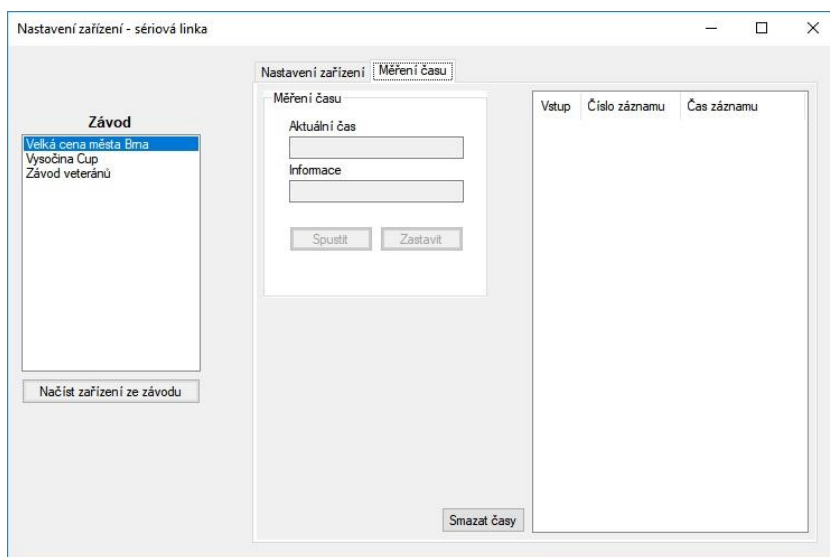
Okno slouží pro nastavení externích zařízení (zařízení pro měření, zobrazovací jednotka). Jak již bylo naznačeno výše, nastavení se značně pozměnilo.

### 5.2.1 Původní vzhled a funkce

Okno bylo rozděleno do dvou podoken (viz Obrázek 5 a Obrázek 6). První podokno umožňovalo nastavit měřicí zařízení až po připojení se k databázi. Zařízení bylo potřeba načíst ze zařízení přidělených danému závodu (toto přidělení se provádělo v jiném okně aplikace). Poté uživatel přiřadil měřicí jednotce COM port počítače a nastavil veškeré parametry pro přenos informace (rychlost přenosu, počet stop bitů, start bitů, handshake, počet datových bitů, parita). Měřicímu zařízení se daly nastavit funkce pro jednotlivé vstupy. Nastavení vstupů mělo umožňovat měření mezičasu kola, funkce ale v programu nebyla dokončena.



Obrázek 5 - Původní nastavení zařízení první podokno



Obrázek 6 - Původní nastavení zařízení druhé podokno

Nastavení měřicího zařízení bylo značně zdlouhavé a podrobné. Uživatel programu se neobešel bez přesných instrukcí.

Druhé podokno sloužilo k testování připojeného měřicího zařízení. Stačilo spustit kontrolní měření a do tabulky se zapisovaly přijaté zprávy.

### 5.2.2 Nový vzhled a funkce

Rozdělení okna do dvou podoken zůstalo stejné, avšak obě podokna se nyní zaměřují na nastavení odlišného zařízení. První podokno slouží pro nastavení zobrazovací jednotky a druhé pro nastavení měřicí jednotky. Vzhled obou podoken je velmi podobný, to zajišťuje snazší práci s programem a orientaci v programu (viz Obrázek 7 a Obrázek 8).

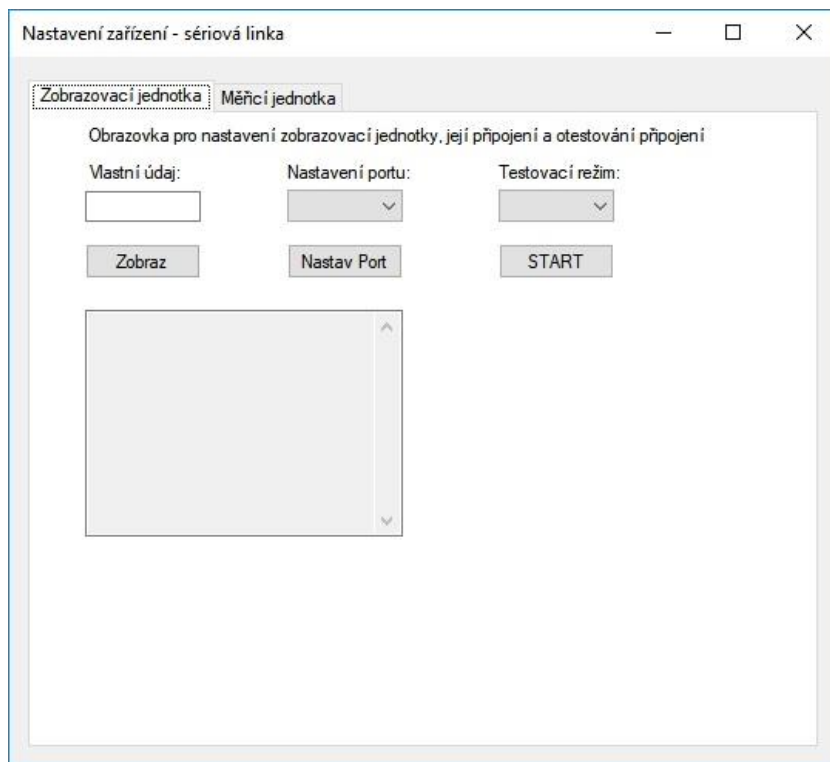
V podokně pro nastavení zobrazovací jednotky se nacházejí tři tlačítka. První tlačítko slouží pro zobrazení jakéhokoliv vlastního údaje na zobrazovací jednotce z textového pole nad tlačítkem (lze zobrazit pouze 4 znaky, více informací viz kapitola 4). Druhé tlačítko slouží pro nastavení příslušného COM portu pro komunikaci s časomírou. COM port se vybírá z rolovací nabídky nad tlačítkem. Poslední (třetí) tlačítko spouští testování zobrazovací jednotky. Z rolovací nabídky nad tlačítkem se vybere testovací režim (Test, Čas).

Testovací režim „Test“ posílá v rozmezí jedné sekundy na zobrazovací jednotku posloupnost číslic od jedné do devíti. Druhý testovací režim „Čas“ posílá na zobrazovací jednotku čas počítače (posílá pouze hodiny a minuty).

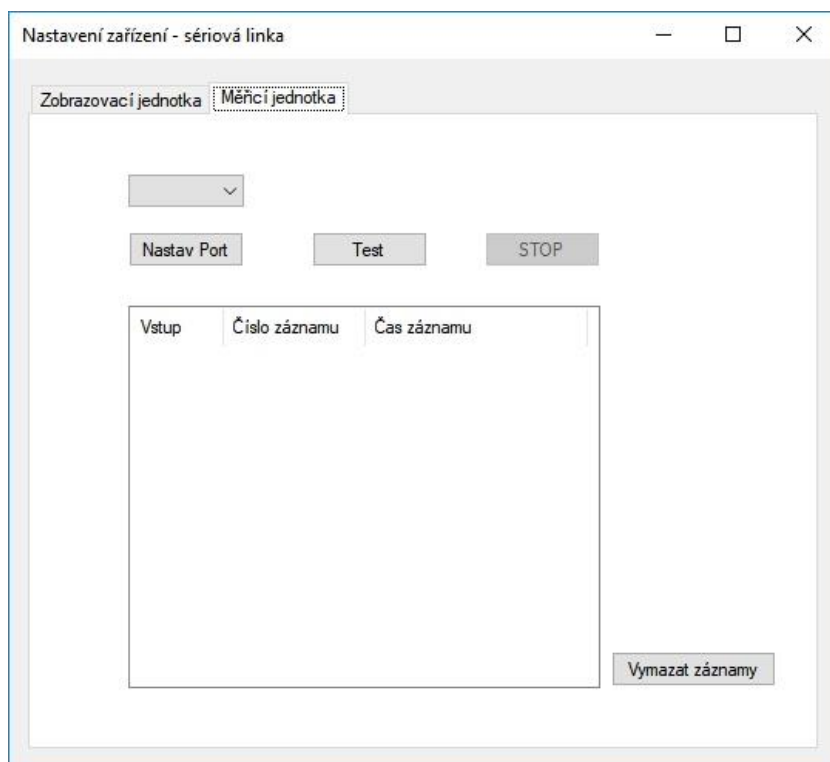
Veškerá poslaná data na zobrazovací jednotku lze vidět v okně pod tlačítky. Toto okno slouží pro kontrolu, aby uživatel aplikace snáze zjistil, kde nastal problém v případě poruchy.

Druhé podokno slouží pro nastavení měřicího zařízení. Nachází se zde jeden rolovací seznam, ze kterého se vybere COM port pro komunikaci, potvrzení probíhá přes tlačítko pod seznamem. Vedle tohoto tlačítka jsou tlačítka „Test“ a „STOP“. Tlačítko „Test“ spouští kontrolní měření, při spuštění měření tlačítko změní barvu na zelenou. Záznamy ze zařízení se zapisují do okna pod tlačítky. Tlačítkem „STOP“ se měření zastaví. Uživatel má také možnost vymazat veškeré naměřené záznamy pomocí tlačítka „Vymazat záznamy“ vpravo dole.

Upustilo se zde od nastavení parametrů připojení (nastaveno pevně v programu). Výhodou je rychlejší nastavení měřicí jednotky a potřeba téměř nulových znalostí pro připojení jednotky. Nevýhodou je možnost připojení pouze jedné měřicí jednotky. Tato nevýhoda by se dala lehce odstranit přidáním rolovacího seznamu, kde by byly vyjmenovány podporované měřicí jednotky. Na základě výběru by se automaticky nastavily parametry přenosu.



**Obrázek 7 - Nové nastavení zařízení první podokno**



**Obrázek 8 - Nové nastavení zařízení druhé podokno**



## 6 PROGRAM PRO KOMENTÁTORA A PADDOCK

U všech sportovních disciplín, ať už motoristických, či atletických, které jsou vysílány například televizním přenosem, je potřeba mít komentátora, který diváky seznámí s tím, co se právě odehrává. Komentátor potřebuje informace o tom, co se děje. Tyto informace by měly být detailnější než ty, které se zobrazí měřiči času. Zároveň komentátor nepotřebuje ovládat měřicí zařízení, tím pádem nepotřebuje mít ve své aplikaci tlačítka pro obsluhu databáze a různých jiných režijních věcí.

Vlastní programové řešení nese název „Regularity Rally Reporter“, název vychází z již existující aplikace pro měření, a to „Regularity Rally“. Slovo „reporter“ z anglického překladu znamená komentátor. Vzhledem k původně zvolenému anglickému názvu autor zvolil anglický název i pro tuto aplikaci.

Program je vytvořen ve vývojovém prostředí Microsoft Visual Studio 2017, pomocí programovacího jazyka C#. Program zvládá komunikaci s databázovým serverem, jedná se převážně o načítání dat, avšak i přidání poznámky k závodníkovi. V základním nastavení program pracuje s lokální databází, ale při nastavení jiného serveru dokáže pracovat i se vzdálenou databází (otestováno pouze v lokální síti LAN).

Základní inspirace pro tento program byla z programu Display Circuit PRO [8], od společnosti TAG HEUER.

### 6.1 Funkce programu

Regularity Rally Reporter slouží pro přehledné zobrazení výsledků závodů pravidelnosti. Uživatel se může podívat jak na výsledky již proběhlých závodů, tak na výsledky právě probíhajících závodů. Program automaticky aktualizuje zobrazovaná data při přidání nového časového údaje pomocí programu Regularity Rally, nebo „manuálně“ v databázi.

Aplikace umožňuje načtení závodů a zobrazení závodníků, kteří se závodu účastní. Na hlavní obrazovce se objeví seznam závodníků a jejich postup v závodě (odjetá kola, ztráta na prvního závodníka, pořadí v závodě atp.). Ke každému závodníkovi je možnost si zobrazit detailnější informace, kde se nachází například informace o voze se kterým závodí, rovněž lze přidat libovolnou poznámku.

Mimo zobrazení závodníků v závodě aplikace umožňuje zobrazení závodníků v turnaji. Po načtení turnaje se na hlavní obrazovce objeví závodníci, kteří se turnaje účastní, jejich bodové hodnocení a pořadí v turnaji.

Za zmínku stojí fakt, že ačkoli se jedná o zobrazení výsledků pro závody pravidelnosti, tak komentátor, nebo paddock dostává informace o probíhajícím závodě stejné, jako by se jednalo o normální závod (vyhrává ten, co dojel jako první). Z toho důvodu je pořadí závodníků počítáno podle nejkratšího času na trati, tato podmínka totiž zaručuje, že je závodník ve vedení. Důvodem této podmínky jsou přísná pravidla závodů pravidelnosti (závodníci u sebe nesmí mít žádné zařízení, které by umožňovalo měřit čas, ani nic podobného) a také to, že referenční kolo bývá často voleno až po skončení závodu.



## 6.2 Třídy v programu

V programu se nachází několik tříd, pomocí kterých je zajištěn chod celého programu. Jedná se o třídy **ConnectMethodsToDB**, **GlobalData**, **GlobalMethods**, **MethodsForAppLogin**, **MethodsForChooseRace**, **MethodsForChooseTour** a **Permission**.

Důvodem rozdělení do tolika tříd byla přehlednost programu a nezáměnnost jednotlivých metod uvnitř těchto tříd. Všechny metody ve třídách jsou implementovány jako *static* metody. Tato implementace metod umožňuje volání jednotlivých metod, aniž by byla vytvořena nová instance objektu dané třídy, pouze pomocí direktivy „.“ (např. *GlobalMethods.getServerName()*).

Více tříd také umožňuje přesun logické části programu z kódu pro grafickou část programu do tříd. Jinými slovy to znamená, že v metodě pro stlačení tlačítka bude jeden řádek kódu, který zavolá metodu z příslušné třídy. Na základě této zvané metody se vykoná potřebná logická část.

### 6.2.1 Třída **ConnectMethodsToDB**

V této třídě se nacházejí metody, pomocí kterých se program připojí k databázovému serveru. Pomocí připojovacího řetězce (obsahuje „jméno“ serveru, název databáze a přihlašovací údaje) a MySQL příkazu se provede daný příkaz na databázovém serveru. Metody dokážou buďto získat data z databázového serveru, nebo přidat data do databáze, popřípadě upravit existující data, nebo data úplně vymazat. Záleží na charakteru MySQL příkazu.

Jedna z hlavních metod v této třídě je metoda *connectDB*, která zprostředkovává jednostrannou komunikaci se serverem, a to směrem k serveru. Pomocí této metody lze přidávat nové záznamy do databáze, upravovat existující záznamy, anebo záznamy zcela vymazat.

Dále se zde nachází metoda *getDBValues*. Metoda umožňuje získat data z databázového serveru, dle příslušného MySQL příkazu, a data naplnit do objektu „Dictionary“ (slovník, jedná se o objekt jazyka C#). Metoda má své přetížení, které umožňuje naplnění slovníku i za předpokladu, že se v načtených datech vyskytují stejné klíče (objekt slovník pro své fungování potřebuje naplnit hodnotu „klíč“, klíč musí být unikátní v celém slovníku, protože na základě něj se vyhledávají data, která danému klíči odpovídají). Tato skutečnost je vyřešena tak, že se na místo klíče ve slovníku zapisuje posloupnost čísel od 1 do X.

Metoda *Logout*, jak již název napovídá, slouží k odhlášení uživatele z databázového serveru. V případě, že je uživatel přihlášen i do aplikace, dojde nejprve k odhlášení uživatele z aplikace. Při volání této metody se provede „vyprázdnění“ všech načtených dat a vymazání přihlašovacích údajů z proměnných v programu.

Všechny metody, které zprostředkovávají přístup k databázovému serveru využívají metodu *CheckConnection*, která slouží k ověření připojení k databázovému serveru. Za předpokladu, že se ověření nezdaří, metoda vypíše chybovou hlášku. Všechny metody ve třídě lze nastavit tak, aby zobrazovaly chybové hlášky v případě poruchy.

Většina metod v této třídě je přejatých z programu Regularity Rally a upraveny dle požadavků programu Regularity Rally Reporter.

## 6.2.2 Třída GlobalData

Tato třída obsahuje proměnné, se kterými je třeba pracovat pro připojení se k databázovému serveru. Konkrétně se jedná o proměnné *login*, *password*, *server* a *database*.

Metody, které tato třída obsahuje, slouží k získání, nebo změně hodnoty výše zmíněných proměnných. Jsou to například tyto metody *SetDatabase* (slouží k nastavení jména databáze), *SetServer* (slouží k nastavení jména serveru), *GetServer* (slouží k získání jména serveru).

Součástí třídy je také metoda *GetConnectionString*, sloužící k získání připojovacího řetězce, pomocí kterého se lze přihlásit k databázovému serveru. Metoda je volána pouze z třídy *ConnectMethodsToDB* popsané nahoře.

## 6.2.3 Třída GlobalMethods

Obsahuje metody pro aktualizaci dat zobrazovaných na hlavní obrazovce aplikace. Jde o metody *UpdateDataListViewRaces* a *UpdateDataListViewTours*.

Metody porovnávají data, která jsou zobrazená na hlavní obrazovce aplikace s uloženými daty na databázovém serveru. V případě, že data sobě neodpovídají, dojde k zavolání příslušných metod pro aktualizaci dat v aplikaci. Porovnávání dat probíhá na základě změny počtu kol u jednotlivých závodníků. Tento způsob byl zvolen z důvodu, že počet kol by měl být nejčastěji měněný údaj. Kontrola všech údajů nedává smysl a při velkém počtu závodníků by se tato operace stala procesně velmi náročnou.

Metody v této třídě jsou periodicky volány pomocí časovače a takto je zajištěna aktuálnost dat v aplikaci.

Třída také obsahuje metody pro získání a zápis názvu databázového serveru a databáze do textového souboru. Ukládání těchto údajů do textového souboru bylo zvoleno z důvodu, aby uživatel při každém spuštění aplikace nemusel tyto údaje zadávat, v případě, že by se nechtěl připojit do lokální databáze (defaultní nastavení).

## 6.2.4 Třída MethodsForAppLogin

Jak již název napovídá, třída obsahuje metody, které jsou potřebné pro přihlášení uživatele do aplikace. Obsahuje proměnné, které uchovávají přihlašovací jméno uživatele a jeho přiřazenou roli. Z důvodu bezpečnosti se neuchovává heslo (to také není potřeba, protože při prvotním přihlášení aplikace zjistí všechna potřebná data, aby byla schopná uživatele později identifikovat).

Metody obsažené v této třídě zprostředkovávají informace o tom, zda je uživatel přihlášen, jaké je přihlašovací jméno uživatele a jakou má přihlášený uživatel přiřazenou roli (role může být komentátor, časoměřič, nebo administrátor). Dále obsahuje metody pro zjištění „ID“ uživatele z databázového serveru, stejně tak jeho role.

Nachází se zde také metoda *LogOut*, která podobně, jako v třídě **ConnectMethodsToDB** slouží k odhlášení. Tentokrát se ale jedná o odhlášení uživatele z aplikace. V případě odhlášení uživatele dojde k vyprázdnění nashromážděných dat o uživateli a k vyprázdnění načtených dat zobrazených závodů, nebo turnajů.

### 6.2.5 Třída **MethodsForChooseRace**

Metody ze třídy jsou využívány při výběru závodu z databáze a k načtení závodníků, kteří v daném závodě závodí. Třída obsahuje dvě hlavní metody *LoadRaces* a *FillRacers*.

*LoadRaces* zajišťuje načtení všech závodů, které se nachází v databázi, nebo v turnaji (v případě, že je nějaký turnaj vybrán). Načtené závody jsou zobrazeny v okně pro výběr závodů. Po výběru závodu je poté volána metoda *FillRacers*.

Metoda *FillRacers* má na starost získání informací o závodnících ve vybraném závodě a zobrazení těchto informací v hlavním okně.

Dále tato třída obsahuje metody pro získání názvu závodu, zjištění ID závodu a také metodu, která indikuje, zda je závod vybrán, či nikoliv.

### 6.2.6 Třída **MethodsForChooseTour**

Tato třída je ve své podstatě stejná jako třída **MethodsForChooseRace** popsaná nahoře. Rozdíl mezi těmito třídami je ten, že třída **MethodsForChooseTour** obsahuje metody pro správu turnajů, nikoliv závodů. Metody mají i obdobné názvy, a to *LoadTours* a *FillRacers*.

### 6.2.7 Třída **Permission**

V třídě *Permission* jsou metody zjišťující, zda má uživatel přístup k požadovaným věcem. To znamená, zda má uživatel právo na zobrazení závodů, nebo turnajů, právo na vložení poznámky k jezdci, nebo ne.

V této třídě se aktuálně nachází pouze jedna metoda a to *CanRead*. Metoda vrací logickou hodnotu „true“, nebo „false“ na základě toho, zda přihlášený uživatel má právo číst záznamy z databáze.

V případě, že se do aplikace přihlásí uživatel s rolí „Komentátor“, tato metoda vrátí vždy „true“, i v případě, že uživatel chce vkládat poznámku k závodníkovi. Metoda je uzpůsobená tomu, aby vyhověla všem požadavkům role „Komentátor“.

## 6.3 Práce s databázovým modelem

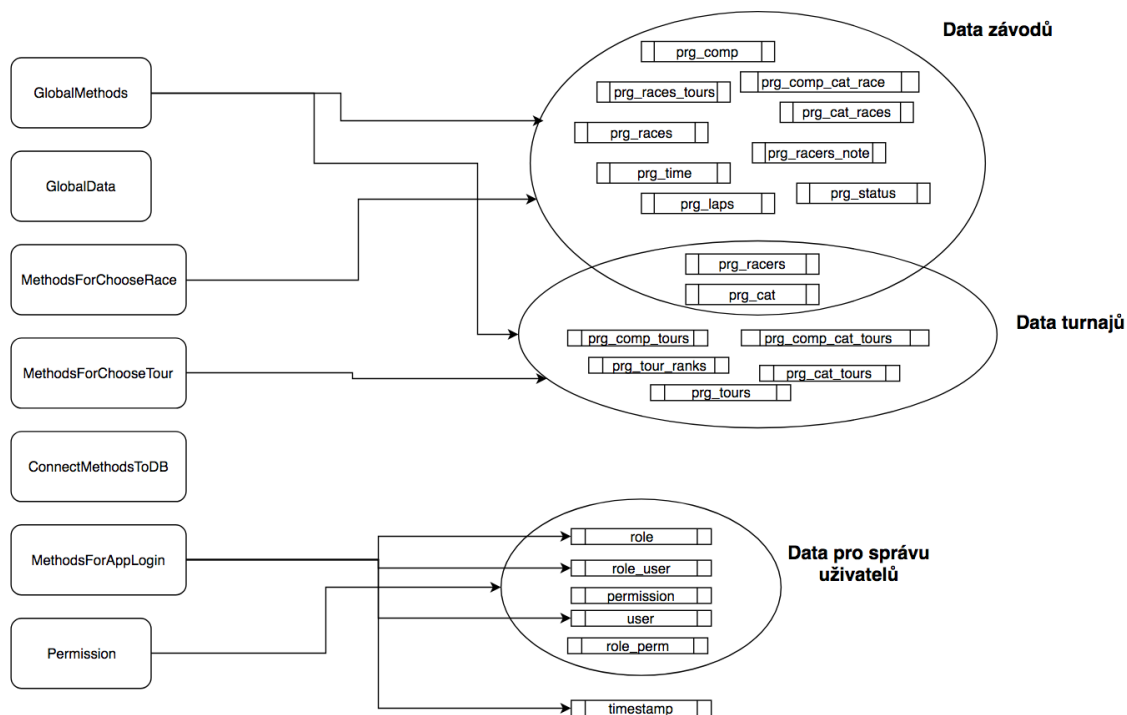
Databázový model vytvořený pro Regularity Rally obsahuje mnoho tabulek a dat (viz PŘÍLOHA 1 – DATABÁZOVÝ MODEL). Aplikace Regularity Rally Reporter nepotřebuje pracovat se všemi daty, které se v databázi nachází. Hlavní data z databáze, která jsou potřeba se týkají čistě závodníků, kteří závodí v daném turnaji, nebo závodě, proto se nepracuje se všemi datovými tabulkami. Výčet všech tabulek se kterými aplikace pracuje se nachází dole.

Tabulka 12 - Databázové tabulky využívané v Regularity Rally Reporter

Název tabulky	Popis tabulky
prg_cars	Informace o vozidle
prg_cat	Informace o závodní kategorii
prg_cat_races	Spojení kategorie se závodem
prg_cat_tour	Spojení kategorie s turnajem
prg_comp	Informace o soutěžících v závodě

prg_comp_cat_race	Přiřazení soutěžícího k závodní kategorii
prg_comp_cat_tour	Přiřazení soutěžícího k turnajové kategorii
prg_comp_tour	Soutěžící v turnaji
prg_laps	Tabulka jednotlivých zajetých kol
prg_racers	Informace o jezdcích
prg_races	Informace o závodních podnicích
prg_races_tour	Přiřazení závodu do turnaje
prg_status	Tabulka statusů
prg_time	Tabulka se závodními časy
prg_tour_ranks	Pořadí soutěžících v turnaji
prg_tours	Informace o turnaji
prg_racers_note	Uchovává poznámky k jednotlivým závodníkům
User	Přihlašovací údaje o uživateli
Role	Názvy jednotlivých rolí
permission	Jednotlivá oprávnění
timestamp	Časová známka
role_user	Přiřazení rolí uživatelům
role_perm	Přiřazení oprávnění rolím

Načítání dat neprobíhá naráz při spuštění dat, ale jednotlivé třídy si načítají nezbytná data pro jejich fungování zvlášť. Diagram znázorňující tuto komunikaci viz Obrázek 10 dole.



Obrázek 10 - Komunikace mezi metodami a databází

## 6.4 Vzhled a funkce jednotlivých programových oken

V následující kapitole budou popsána jednotlivá okna aplikace a jejich funkce. Popis jednotlivých programových oken bude následovat posloupnost, v jaké se uživatelé zobrazují při spuštění aplikace.

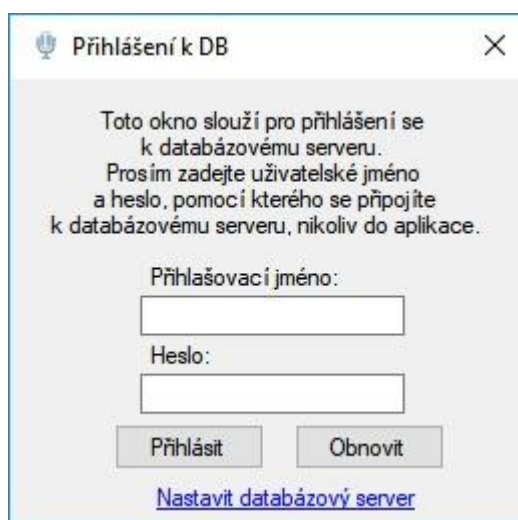
### 6.4.1 Přihlášení do databáze

Prvním oknem, které se uživateli zobrazí, je požadavek na přihlášení do databáze. V tomto okně uživatel zadá uživatelské jméno a heslo, které má vytvořené v databázovém serveru.

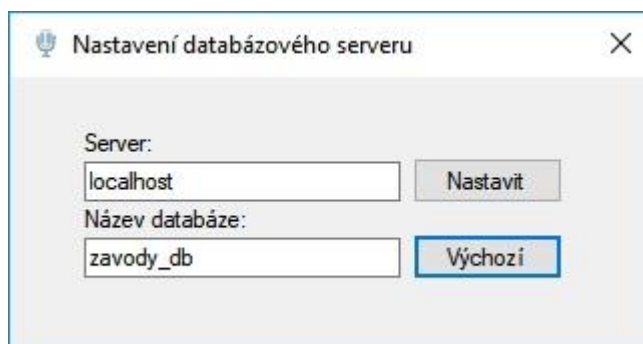
V případě, že se databáze nenachází na stejném PC, na kterém je spuštěna i aplikace Regularity Rally Reporter, je třeba nastavit databázový server, ke kterému se má aplikace připojit (implicitně je databázový server nastaven na *localhost*). Také si uživatel může změnit název databáze, v případě, že vytvořil databázi pod jiným názvem, jak defaultně nastaveným (*zavody\_db*). Skript pro vytvoření databáze automaticky nastavuje jméno databáze na *zavody\_db*.

Při neúspěšném přihlášení se zobrazí chybová hláška s popsáním problému, kde nastala chyba (špatné přihlašovací jméno, špatné heslo, server nenalezen atp.).

Po úspěšném přihlášení se zobrazí informační okno, které potvrzuje přihlášení.



Obrázek 11 – Okno pro přihlášení se k databázovému serveru

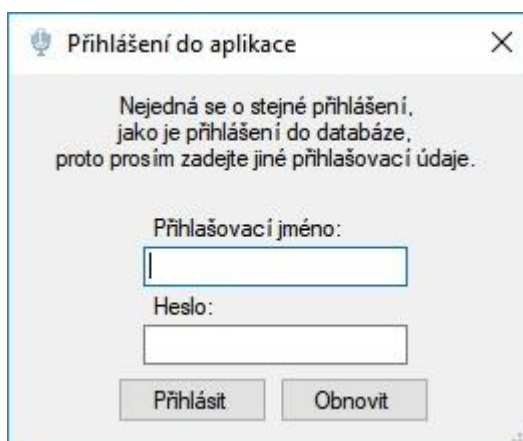


Obrázek 12 – Okno pro nastavení databázového serveru

## 6.4.2 Přihlášení do aplikace

Po úspěšném přihlášení do databáze se objeví okno, pomocí kterého se uživatel přihlásí do aplikace. Jedná se o jiné údaje než ty, pomocí kterých se dá připojit k databázovému serveru.

Toto přihlášení slouží pro jednotlivé uživatele, kteří mohou aplikaci používat. Každý uživatel má také svou přiřazenou roli. V aplikaci se zatím nacházejí pouze tři role (komentátor, administrátor a časoměřič). Časoměřič nemá přístup do této aplikace. V případě, že se přihlásí uživatel s rolí časoměřiče, bude jeho pokus o přihlášení odmítnut. Další dvě role (administrátor a komentátor) mají přístup do tohoto programu.



Obrázek 13 – Okno pro přihlášení se do aplikace

## 6.4.3 Hlavní programové okno

Po přihlášení se do databázového serveru a do aplikace dojde k otevření hlavního programového okna. Do tohoto okna se uživatel dokáže dostat i v případě, že se nepřihlásí ani přes jedno přihlašovací okno, nebude mu však umožněno nijak s aplikací pracovat.

V hlavním okně se zobrazuje kompletní přehled všech závodníků ve vybraném závodě (horní tabulka), nebo turnaji (spodní tabulka). V horní části aplikace se nachází kontextové menu, ve kterém uživatel nalezne tři políčka (menu, přihlášení a zobrazit). V případě úspěšného připojení do aplikace se vedle tlačítek kontextového menu zobrazuje i přihlašovací jméno uživatele a jeho přiřazená role (viz Obrázek 14).

Pod položkou „menu“ se skrývají další dvě tlačítka (Výběr závodu a Výběr turnaje). Na základě volby uživatele se objeví buďto okno pro výběr závodu, nebo turnaje. V tomto podokně si poté uživatel zvolí příslušný závod nebo turnaj a hlavní okno se naplní hodnotami vybraného závodu či turnaje.

Položka „přihlášení“ skrývá přihlašovací okna, které se automaticky zobrazí při startu aplikace. Uživateli je zde umožněno se odhlásit jak z aplikace, tak i z databázového serveru a opět se znovu přihlásit. Při odhlášení jsou vymazána všechna nashromážděná data a hlavní okno aplikace bude opět prázdné. Skrytí všech dat bylo zvoleno z důvodu ochrany dat.

Poslední položkou kontextového menu je tlačítko „zobrazit“. Tato nabídka umožňuje uživateli přepínat mezi třemi typy zobrazení. Buďto se budou zobrazovat data



pouze pro závody, nebo se budou zobrazovat data jen pro turnaje, anebo se zobrazí data jak pro turnaje, tak i závody (implicitní nastavení).

Při vybrání závodu, nebo turnaje se v hlavním okně také zobrazuje název závodu, nebo turnaje, který byl zvolen. U závodu se zobrazuje i jeho stav (probíhá, zrušen, přerušen atp.). Po vybrání závodu, nebo turnaje se zapíná automatická aktualizace dat.

V pravém dolním rohu se nacházejí tři tlačítka, první tlačítko slouží pro přidání poznámky k jezdcí, druhé zobrazí detail jezdce a poslední tlačítko ruší výběr turnaje. Tlačítka pro přidání poznámky a zobrazení detailu fungují pouze pro závody a stávají se aktivními při výběru jezdce z tabulky jezdců v závodě (horní tabulka). Tlačítko pro zrušení výběru turnaje se stává aktivní po vybrání turnaje.

Hlavní programové okno je plně responzivní na velikost okna, omezen je uživatel pouze nejmenší velikostí okna (1120 x 730 pixelů). Tato velikost je zvolena z toho důvodu, aby si hlavní programové okno udrželo určitou čitelnost a přehlednost.

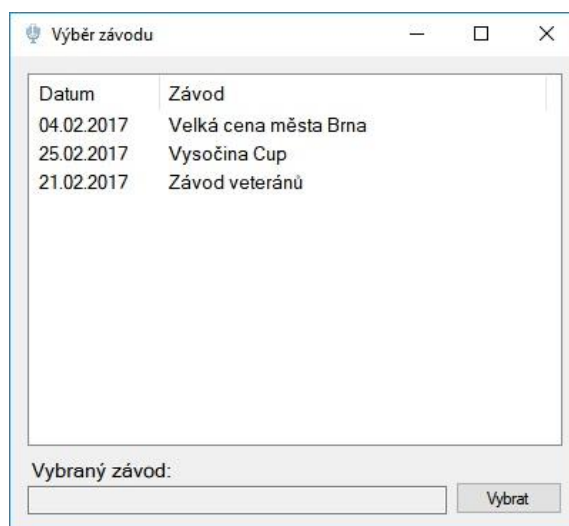
**Obrázek 14 – Hlavní okno programu Regularity Rally Reporter s načtenými daty**

### 6.4.4 Okno pro výběr závodu a výběr turnaje

Okno pro výběr závodu má za úkol výběr závodu a zobrazení jednotlivých závodníků, kteří se závodu účastní, v hlavním okně programu. Do výběru závodu se uživatel dostane pouze poté, co je přihlášen do aplikace a role uživatele splňuje potřebná práva pro zobrazení závodů.

Při načtení okna pro výběr závodu se načtou všechny dostupné závody do tabulky a uživateli stačí vybrat, který závod chce zobrazit (viz Obrázek 15). Po označení závodu se výběr zobrazí v okně pod tabulkou, poté stačí buďto kliknout na tlačítko „vybrat“, nebo dvakrát kliknout na závod v tabulce.

Výběr turnaje probíhá zcela stejně jako výběr závodu, proto zde nebude popisován. Jediná věc, co stojí za zmínku je, že při výběru turnaje se zúží okruh vybíraných závodů. V případě, že je vybrán turnaj, tak uživatel bude moci vybrat pouze závody, které se nacházejí v daném turnaji. Výběr turnaje lze zrušit pomocí tlačítka v hlavním programovém okně.



Obrázek 15 – Okno pro výběr závodu

### 6.4.5 Detailní okna aplikace

V aplikaci se nachází dvě detailnější okna. Jedná se o zobrazení detailu jezdce a přidání poznámky k jezdci. Přístup k těmto oknům byl vysvětlen v kapitole 6.4.3.

Okno pro přidání poznámky zobrazuje jen startovní číslo vybraného závodníka a jeho jméno. Poznámku lze vepsat do okna určeného pro poznámku, stačí jen kliknout na volné pole a začít psát. Při kliknutí na tlačítko „uložit“ se poznámka uloží do databáze a dojde k uzavření okna pro přidání poznámky. Když uživatel klikne na tlačítko „smazat“, poznámka se smaže, úpravy se uloží do databáze a okno se opět zavře. V případě, že uživatel napíše k jezdci poznámku a poznámku neuloží, bude při zavření okna dotázán, zda chce pokračovat bez uložení, nebo chce poznámku uložit.

<b>Jméno:</b>	Petr Novák
<b>Číslo:</b>	3
<b>Poznámka:</b>	Poznámka u jezdce

Uložit Vymazat

Obrázek 16 – Okno pro přidání poznámky

Okno detail jezdce obsahuje podrobnější informace o zvoleném jezdci, neobsahuje žádná editovatelná okna. Je zbytečné zde popisovat, co vše toto okno zobrazuje, lepší vypovídací hodnotu má obrázek (viz Obrázek 17).

<b>Jméno:</b>	Petr Novák
<b>Číslo:</b>	3
<b>Pohlaví:</b>	Muž
<b>Datum naroze...</b>	11.12.1968 0:00:00
<b>Národnost:</b>	CZE
<b>Auto:</b>	Škoda Felicia 2004
<b>Tým:</b>	Novak Team
<b>Poznámka:</b>	Poznámka u jezdce

Obrázek 17 – Okno detail jezdce

## 7 ZÁVĚR

Tato bakalářská práce je zaměřená na závody pravidelnosti. V úvodu práce je jednoduchý popis této závodní disciplíny. Úkolem práce bylo sestavit modul pro komunikaci s časomírou a zabudovat jej do programu Regularity Rally (viz kapitola 5). Jedná se o komunikaci na zcela základní úrovni, a to posílání dat směrem k zobrazovací jednotce. Pomocí modulu lze na zobrazovací jednotce zobrazit jakákoliv data (v rámci možností 7 segmentového zobrazovače), nebo využít jeden z módů zobrazení výsledků. Do budoucna by modul mohl umožňovat zobrazení více údajů, ale to by znamenalo i vylepšení zobrazovací jednotky. Modul pro komunikaci je popsán v kapitole 4.

Dále je v práci popsán vývoj aplikace pro komentátora a paddock. Aplikace získává data ze stejné databáze, která byla použita pro Regularity Rally. Databáze se dostala pouze malých změn, které jsou popsány v kapitole 3. Aplikace umí pracovat s databázovými tabulkami pro správu uživatelů, dokáže tak zamezit přístupu do aplikace uživateli, který nemá povoleno s touto aplikací pracovat.

Nedílnou součástí této práce je také oprava programu Regularity Rally. Prvně bylo potřeba se se stávajícím programovým řešením seznámit a najít chyby, které je třeba opravit. Seznámení se s tímto programem je popsáno v kapitole 2. Na základě zjištěných nedostatků a dalších podnětů od vedoucího bakalářské práce bylo opraveno několik zásadních chyb v aplikaci (viz kapitola 5).

V aplikaci bylo opraveno několik zásadních chyb. Upravilo se měření závodníků. Nyní je umožněno rychlejší měření a možnost měření více časů v krátkém časovém okamžiku (viz kapitola 5.1). Dále bylo upraveno nastavení měřicí jednotky, které se značně zjednodušilo (viz kapitola 5.2).

Mnoho cenných rad a přístupů k dané problematice bylo nalezeno na webovém serveru Stackoverflow (<https://stackoverflow.com>). Jedná se o webové stránky (fórum) zaměřené na programování. Těchto webových stránek bylo převážně využíváno při řešení problémech v kódu, se kterými si autor práce nevěděl rady.

Autorovi tato bakalářská práce byla jednoznačným přínosem v oblasti jazyka C#, vývoji aplikací a také v databázových systémech. Autor práce neměl žádné zkušenosti ani v jedné z těchto oblastí před započatím práce na bakalářské práci. Při osvojování jazyka C# autor využil znalostí získaných studiem jazyka C++ na vysoké škole.

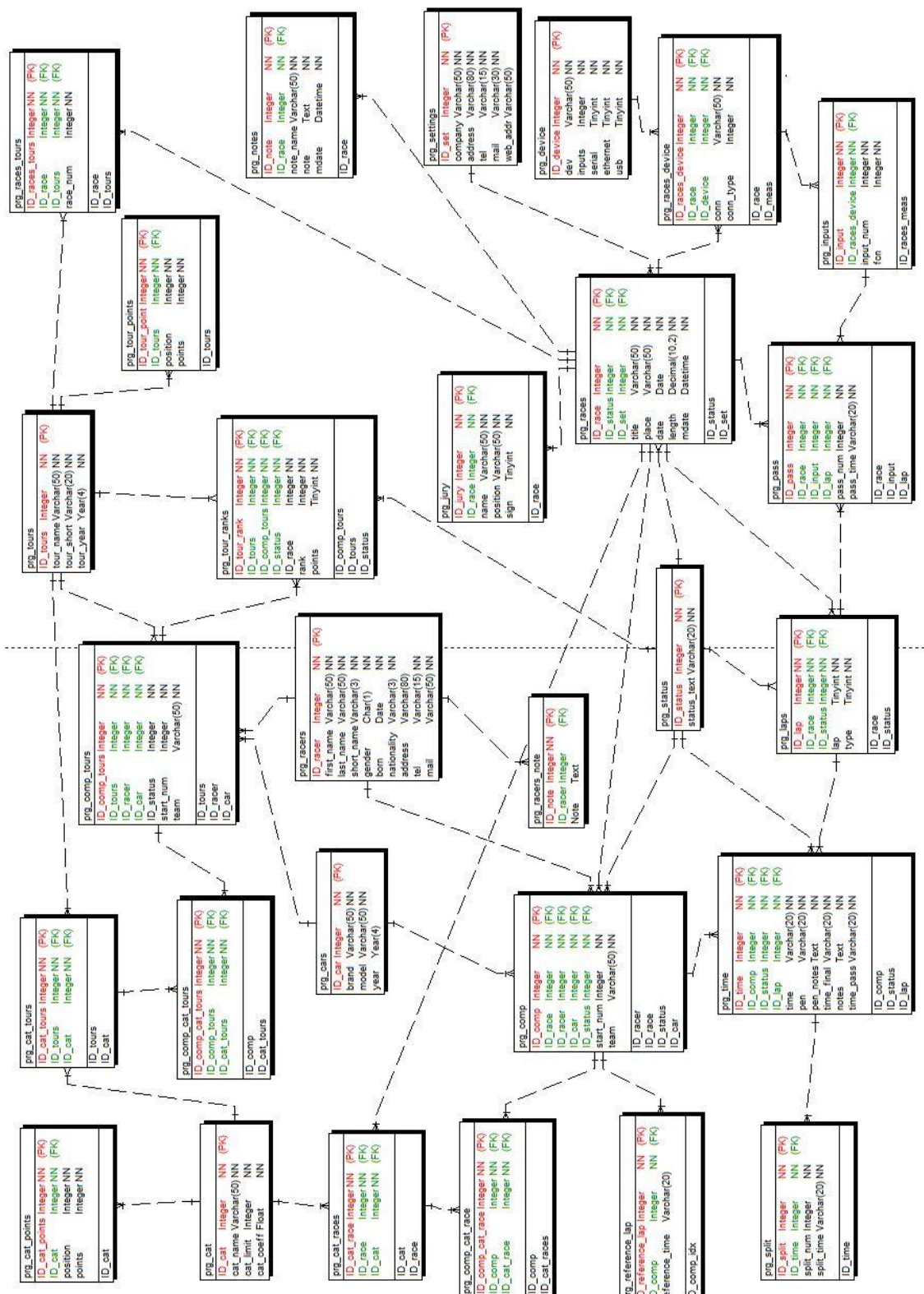
# Literatura

- [1] TAG-HEUER CHRONOPRINTER 505...QUICK OPERATING GUIDE [online]. Reliable Racing Supply, 1992 [cit. 10.05.2018]. Dostupné z: <http://www.reliableracing.com/downloads/505man.pdf>
- [2] Protocol for HL960/HL990 [online]. TAG HEUER PROFESSIONAL TIMING. Switzerland [cit.2018-01-05]. Dostupné z: <http://www.pyramidthiming.com/Documentation/HL%20960%20protocol.pdf>
- [3] KŘÍŽ, P. Software pro jízdu pravidelnosti. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 91s. Vedoucí diplomové práce Ing. Radek Štohl, PhD
- [4] Download MySQL Workbench [online]. ORACLE. [cit. 10.05.2018]. Dostupné z: <https://dev.mysql.com/downloads/workbench/>
- [5] TAMBI, Yash. Serial Communication – Introduction [online]. VIT UNIVERSITY. Vellore, 3.9.2013 [cit. 10.05.2018]. Dostupné z: <http://maxembedded.com/2013/09/serial-communication-introduction/>
- [6] Vola Timing: Circuit-Pro Suite. Vola Timing. [online]. [cit. 10.05.2018]. Dostupné z: <http://www.vola.fr/en/timing/logiciels/suite-circuit/suite-circuit-live>
- [7] MySQL. MySQL. [online]. [cit. 10.05.2018]. Dostupné z: <http://www.mysql.com/>
- [8] TAG Heuer: Display Circuit PRO [online]. [cit. 10.05.2018] Dostupné z: <http://www.tagheuer-timing.com/en/tag-heuer-professional-display-circuit-pro>
- [9] HHD Software: Device Monitoring Studio [online]. [cit. 10.05.2018] Dostupné z: <https://www.hhdsoftware.com/Downloads/device-monitoring-studio>
- [10] VEČEŘA, J. Scoreboard pro časomíru. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 51 s. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D.
- [11] HEVES, R. Scoreboard pro časomíru motoristických podniků. BRNO: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2007. 60 s. Vedoucí diplomové práce Ing. Radek Štohl, Ph.D.
- [12] LICHOSYT, T. Software pro jízdu do vrchu. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 82s. Vedoucí diplomové práce Ing. Radek Štohl, PhD.

# Seznam příloh

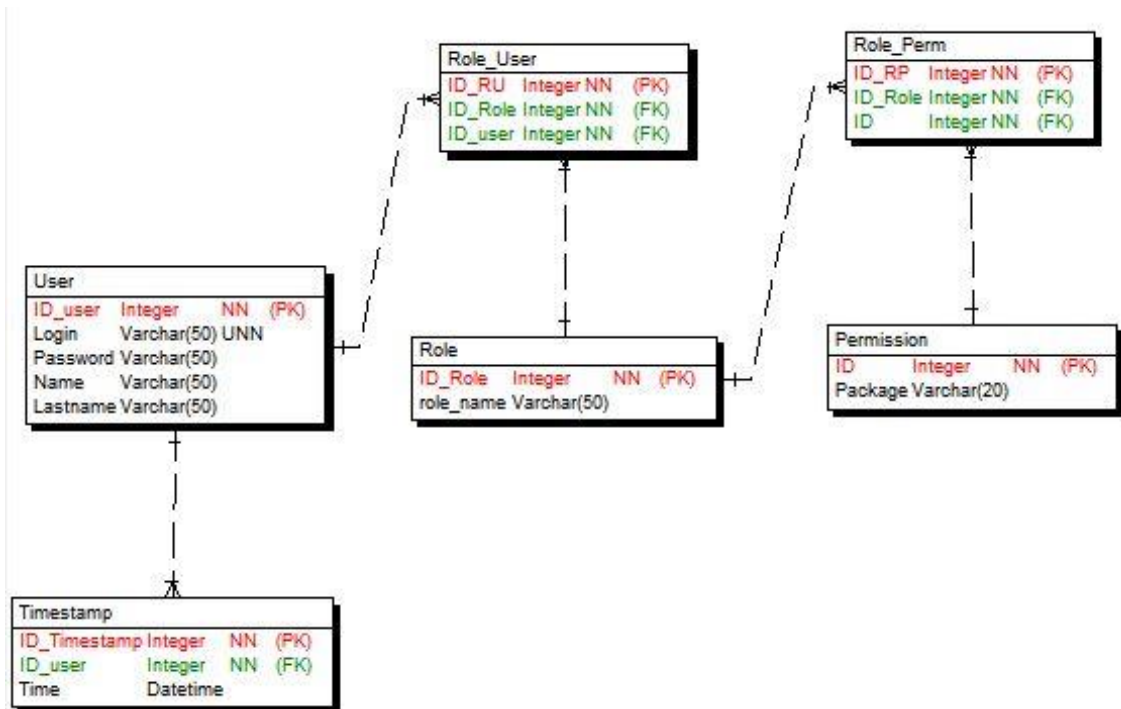
1. Databázový model
2. Databázový mode – správa uživatelů
3. Přiložené CD/DVD

## Příloha 1 – Databázový model





## Příloha 2 – Databázový model – správa uživatelů



## **Příloha 3 – CD/DVD**

Na přiloženém CD/DVD se nacházejí následující data:

- Elektronickou verzi bakalářské práce ve formátu PDF
- Zdrojové soubory pro aplikaci Regularity Rally
- Instalační soubory aplikace Regularity Rally
- Zdrojové soubory pro aplikace Regularity Rally Reporter
- Instalační soubory aplikace Regularity Rally Reporter
- Soubor s SQL skriptem pro vytvoření databázového modelu
- Soubor s SQL skriptem pro vytvoření databázového modelu s ukázkovými daty
- Textový soubor se jmény vytvořených uživatelů a jejich hesly